Nathan Lam, 400070248, Feb 1, 2019

module tail\_lights(

input reset,

left,

right,

clock

output reg LA,

LB,

LC,

RA,

RB,

RC

);

wire clock, reset, left, right;

reg state;

reg clkreset, pulse;

always @(\*)

begin

clock\_divider(clock,clkreset,10'd4,clkout);

if (reset == 1'b1) begin

state <= reset;

end else

case(state)

idle:

if (left == 1'b1) begin

state <= signal\_left;

end else if (right == 1'b1) begin

state <= signal\_right;

end else if (reset == 1'b1) begin

state <= reset;

else state <= idle;

end

signal\_left:

LA <= 1

//pause

@(posedge clkout)

LB <= 1

//pause

@(posedge clkout)

LC <=1

//pause

@(posedge clkout)

LA <=0

LB <=0

LC <=0;

state <= idle;

signal\_right:

RA <= 1

//pause

@(posedge clkout)

RB <= 1

//pause

@(posedge clkout)

RC <= 1

//pause

@(posedge clkout)

RA <=0

RB <=0

RC <=0;

state <= idle;

reset:

LA <=0

LB <=0

LC <=0

RA <=0

RB <=0

RC <=0;

state <= idle;

default:

LA <=0

LB <=0

LC <=0

RA <=0

RB <=0

RC <=0;

state <= idle;

endcase

end

endmodule

module clock\_divider(

input Clock, Reset\_n,

input [20:0] divisor,

output clk\_ms

);

reg [20:0] cntr;

always @(posedge Clock)

begin

if ((Reset\_n==1'b0)||(cntr >= (divisor-1'b1))) cntr[20:0] <= 21'd0;

else cntr[20:0] <= cntr[20:0] + 21'd1;

end

assign clk\_ms = (cntr[20:0]>=(divisor/2))?1'b1:1'b0;

endmodule